Chicago Crime Assessment

Group No - P02

Shashank Shekhar (sshekha4)
Harshit Badiyani (hbbadiya)
Nimisha Tripathi (ntripat)
Satanik Ray (sray7)

Introduction & Related Work

In the past:

- **A number of studies have been conducted to predict the occurrence of crime rate.**
- **These studies have used temporal data to study the crime prediction rate but they have faced difficulty in discovering the nonlinear relationships and dependencies between data.**
- **The main reason for not being able to discover proper relationship in crime rates is the lack of inclusion of the factors governing the crimes.**

Present Work and Motivation:

- **An important motivation to work on this project has been to demystify the role that social-economic factors play in crime prediction. Initially crime rates were thought to only depend on temporal factors. Here we work on combining Social-Economic Factors with temporal and spatial data to improve on the crime prediction rate. We have referenced ”Prediction of Crime Occurrence from MultiModel Data using Deep Learning” by Hyeon-WooKang, Hang-Bong Kang in order to successfully complete our work.**

Problem Statement

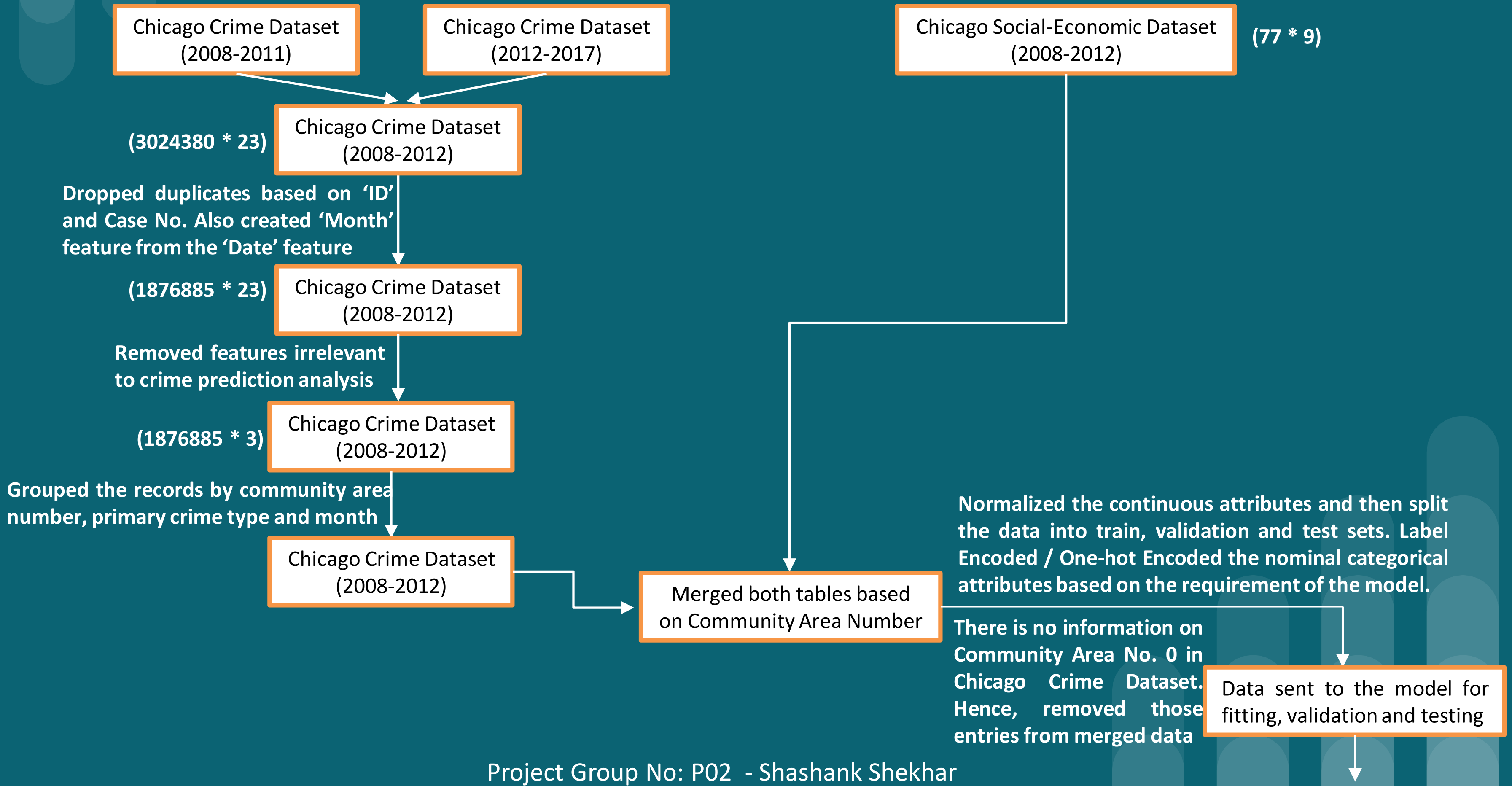
Given a set of Social Economic Factors, develop a model that can accurately predict the number of crimes for a given Community Area, Primary Crime Type and Month

Method

Two main steps:

- Data Preprocessing
- Model Fitting and Hyperparameter Tuning

Data Preprocessing


Chicago Crime Dataset (2008-2011)
Chicago Crime Dataset (2012-2017)
Chicago Social-Economic Dataset (2008-2012)
(77 * 9)
(3024380 * 23)
Chicago Crime Dataset (2008-2012)
(1876885 * 23)
Chicago Crime Dataset (2008-2012)
(1876885 * 3)
Chicago Crime Dataset (2008-2012)
Chicago Crime Dataset (2008-2012)
Merged both tables based on Community Area Number
Data sent to the model for fitting, validation and testing

Dropped duplicates based on 'ID' and Case No. Also created 'Month' feature from the 'Date' feature

Removed features irrelevant to crime prediction analysis

Grouped the records by community area number, primary crime type and month

Normalized the continuous attributes and then split the data into train, validation and test sets. Label Encoded / One-hot Encoded the nominal categorical attributes based on the requirement of the model.

There is no information on Community Area No. 0 in Chicago Crime Dataset. Hence, removed those entries from merged data

Model Fitting and Hyperparameter Tuning

The following models were trained with various hyperparameters:

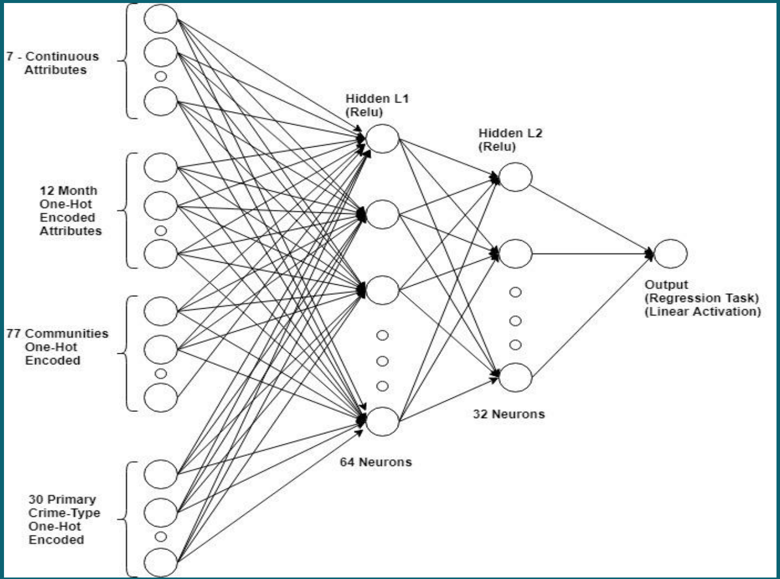
- Multilayer Perceptron -> For this the categorical attributes were one-hot encoded and then passed to the model along with other continuous attributes.

- Random Forest -> For this the categorical attributes were label encoded and then passed to the model along with other continuous attributes.

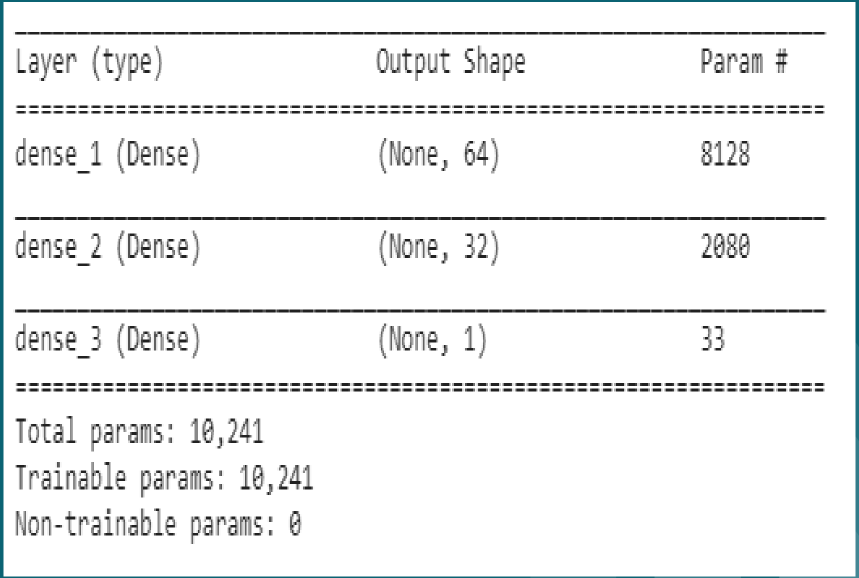
- Support Vector Regressor -> For this the categorical attributes were one-hot encoded and then passed to the model along with other continuous attributes.

Model Fitting and Hyperparameter Tuning

Structure of Multilayer Perceptron Model


7 - Continuous Attributes
12 Month One-Hot Encoded Attributes
77 Communities One-Hot Encoded
30 Primary Crime-Type One-Hot Encoded
Hidden L1 (Relu)
64 Neurons
Hidden L2 (Relu)
32 Neurons
Output (Regression Task) (Linear Activation)

Shape of Layers in Multilayer Perceptron Model



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 64)                8128
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 10,241
Trainable params: 10,241
Non-trainable params: 0
```

The number of layers to be used for this problem have been taken by referencing the paper "Understanding Deep Learning Requires rethinking generalization" by Chiyuan Zhang, Samy Bengio, Moritz Hardt. Benjamin Recht and Oriol Vinyals

Hyperparameter tuning for Multilayer Perceptron Model

Below is the table representing Mean Squared Error for various hyperparameters chosen.

Batch Size: 32

Neurons / Dropout Rate	0.3	0.5	0.8
32	81.5677	122.209	431.065
64	60.3466	93.916	191.587
128	55.285	59.3568	118.251

Batch Size: 64

Neurons / Dropout Rate	0.3	0.5	0.8
32	124.43	133.675	477.695
64	55.3656	72.1274	219.454
128	54.112	58.3809	100.683

Batch Size: 128

Neurons / Dropout Rate	0.3	0.5	0.8
32	110.418	148.159	496.428
64	60.373	106.565	207.229
128	52.5242	65.4541	120.79

Observation from the 3 tables:

Better results are obtained for Higher Batch Sizes, Higher Number of Neurons per Layer and Lower Dropout Rates.

On searching with a batch size = 256, neurons/layer = 256, no dropout, we got our best result of mean squared error of 50.4744 on the validation dataset.

Hyperparameter tuning for Random Forest Model

Below is the table representing Mean Squared Error for various hyperparameters chosen.

n_estimators / max_depth	10	20	30	40
10	130.231	67.3674	64.0749	70.5069
25	120.309	69.414	66.7503	64.4874
50	115.966	61.7925	65.4444	63.1005
100	118.751	64.8371	63.0328	63.609

Observation from the above table:

From the above table it is observed that this model performs comparable to the MLP based model but is not as good as the MLP based model. The best model obtained here is with n_estimators = 50 and max_depth = 20.

Hyperparameter tuning for Support Vector Regressor Model

Below is the table representing Mean Squared Error for various hyperparameters chosen.

Kernel Function: RBF

C / Gamma	0.1	0.001	0.001
0.1	1812.37	1961.95	1988.29
1	1365.06	1730.13	1960.42
10	1065.45	1307.83	1719.95
100	598.666	1140.6	1303.68

Kernel Function: Linear, Gamma: auto
Kernel Function: Polynomial, Gamma: scale

Kernel Function / C	0.1	1	10	100
Linear	1404.0234	1182.4	1148.34	1143.39
Poly	1992.2831	1992.13	1990.6	1977.87

Observation from the above table:

From the above table it is observed that the best support vector regressor model obtained does not perform very well. In general, SVR models did not give good result for our dataset.

Careful Considerations while building the Models

1. **Model and Data Overfitting**

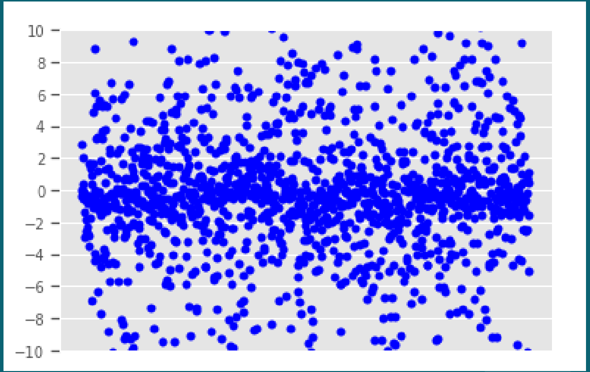
   - For the MLP model, this was taken care of by the dropout layer. During the process of Hyperparameter Tuning, the dropout layer was used. Even after using the dropout layers, it was still observed that the best model was obtained without using any Dropout indicating that the model fits the data well.

   - For the Random Forest Model, the R-squared and adjusted R-squared scores were calculated and were found to be 0.9584 and 0.9345 respectively. The higher values of R-squared and adjusted R-squared indicate that most of the variance in data is explained by the model. The scatter plot of the residuals on the right indicates that this model fits the data well.

   - The value of R-squared score for the Support Vector Regressor Model was very low and hence the model was dropped from further consideration.

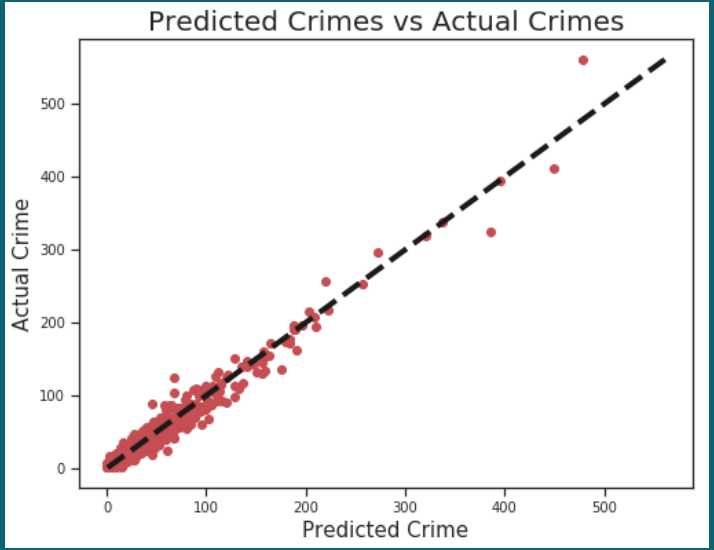
2. **Metrics used to evaluate the models**

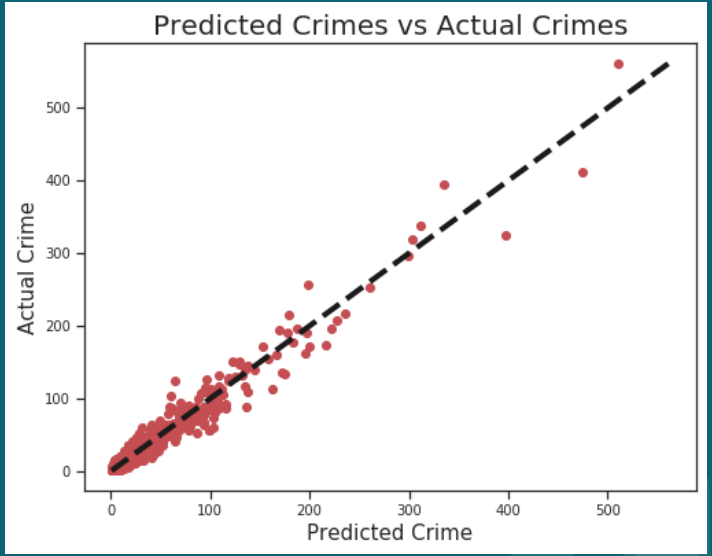
   - Since this is a regression problem and there are no outliers, MSE and RMSE were taken as the metric to evaluate the model.



Scatter plot of residuals

Results

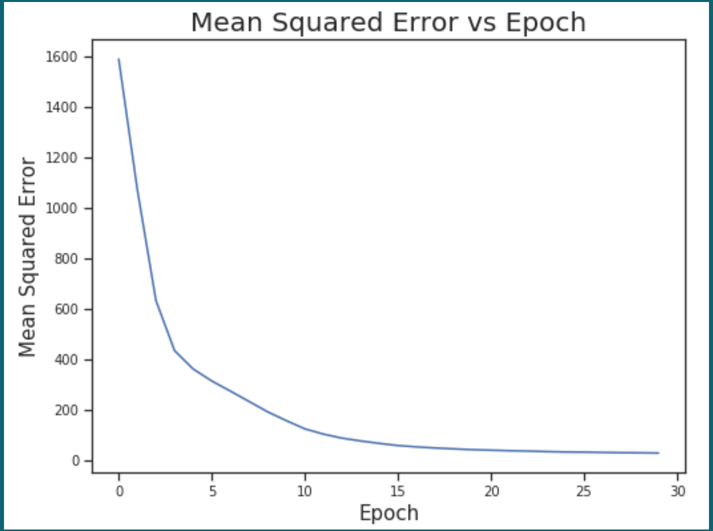




Regression Output: MLP

Regression Output: Random Forest

Results



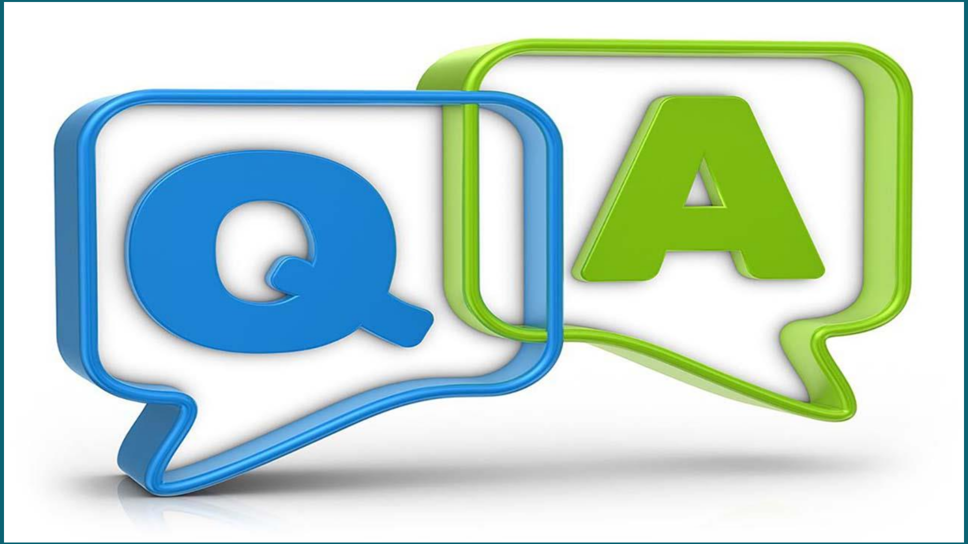
The above graph shows the training curve for the best MLP model. After around 15 epochs, the model is trained and the Mean Squared Error settles to approximately 50.47

Hence, the best model we got is the Multilayer Perceptron Model with a batch size of 256, neurons/layer = 256 without any dropout, we got our best result of mean squared error of 50.4744 (RMSE = 7.10) on the test dataset.

Hence, given a set of Social Economic Factors for the city of Chicago, this model can accurately predict the crimes for a given Community Area, Primary Crime Type and Month.

Discussion and Future Work

- In order to improve on the current best model, Stacking concept could be used to improve performance. The limitations faced to implement this are related to the compatibility of keras with scikit-learn ie. Stacking the keras based MLP implementation with scikit-learn based Random Forest implementation.
- Current Model can also be improved by including additional factors that play an important role in crime prediction. These include Demographics, Weather and other Environmental Factors.
- Some of the features from the original data like latitude and longitude can be used in conjunction with maps to provide the public with the safest travel routes during specific time of day.
- The same information can be shared with the Police Department which can help in proper deployment of cops at different places within the city at specific times of day.



Project Group No: P02